

**Câu 1: Mô tả bằng dạng Bảng và ngôn ngữ tự nhiên các chức năng tương ứng với các actor và phi chức năng của Hệ thống ecomSys**

**1.1. Mô tả hệ thống bằng ngôn ngữ tự nhiên**

- Hệ thống ecomSys là một nền tảng trực tuyến cho phép người dùng mua và bán hàng hóa dịch vụ qua internet. Hệ thống cung cấp một giao diện dễ sử dụng cho người dùng để tìm kiếm các sản phẩm theo nhu cầu. Người dùng có thể truy cập vào hệ thống từ bất kỳ thiết bị kết nối internet nào. Mỗi sản phẩm trên hệ thống được hiển thị với thông tin chi tiết bao gồm hình ảnh, mô tả, giá cả và thông tin giao hàng. Khách hàng có thể đăng ký tài khoản, đăng nhập, xem thông tin sản phẩm, thêm sản phẩm vào giỏ hàng và tiến hành thanh toán trực tuyến hoặc thanh toán khi nhận hàng.

**1.2. Chức năng cho Admin**

- Đây là những chức năng được thực hiện bởi người dùng quản trị.
  - Đăng nhập dành cho admin
  - Quên mật khẩu dành cho admin
  - Chỉnh sửa hồ sơ cho admin
  - Đổi mật khẩu cho admin
  - Chức năng đăng xuất
  - Quản lý khách hàng
    - + Thêm khách hàng mới
    - + Chỉnh sửa khách hàng
    - + Xem thông tin chi tiết của Khách hàng
    - + Danh sách tất cả khách hàng - Quản lý sách
    - + Thêm sách mới
    - + Chỉnh sửa sách đã có
    - + Xem chi tiết sách
    - + Danh sách tất cả sách
  - Quản lý clothes
    - + Thêm mặt hàng quần áo mới
    - + Chỉnh sửa thông tin mặt hàng quần áo
    - + Xem thông tin chi tiết của mặt hàng quần áo
    - + Danh sách tất cả mặt hàng quần áo
  - Quản lý điện thoại
    - + Thêm điện thoại mới
    - + Chỉnh sửa điện thoại đã có trên hệ thống
    - + Xem thông tin chi tiết của điện thoại
    - + Danh sách tất cả điện thoại
  - Quản lý danh mục sách
    - + Thêm danh mục sách mới
    - + Chỉnh sửa danh mục sách đang thoát
    - + Xem chi tiết Danh mục sách
    - + Danh sách tất cả các loại sách
  - Quản lý danh mục quần áo

- + Thêm danh mục quần áo mới
- + Chỉnh sửa danh mục quần áo đang thoát
- + Xem chi tiết Danh mục quần áo
- + Danh sách tất cả các loại quần áo
- Quản lý danh mục điện thoại
  - + Thêm danh mục điện thoại mới
  - + Chỉnh sửa danh mục điện thoại đang thoát
  - + Xem chi tiết Danh mục điện thoại
  - + Danh sách tất cả các loại điện thoại
- Quản lý đơn hàng
  - + Tạo đơn hàng mới
  - + Chỉnh sửa đơn hàng đã có
  - + Xem thông tin chi tiết của đơn hàng
  - + Danh sách tất cả các đơn hàng

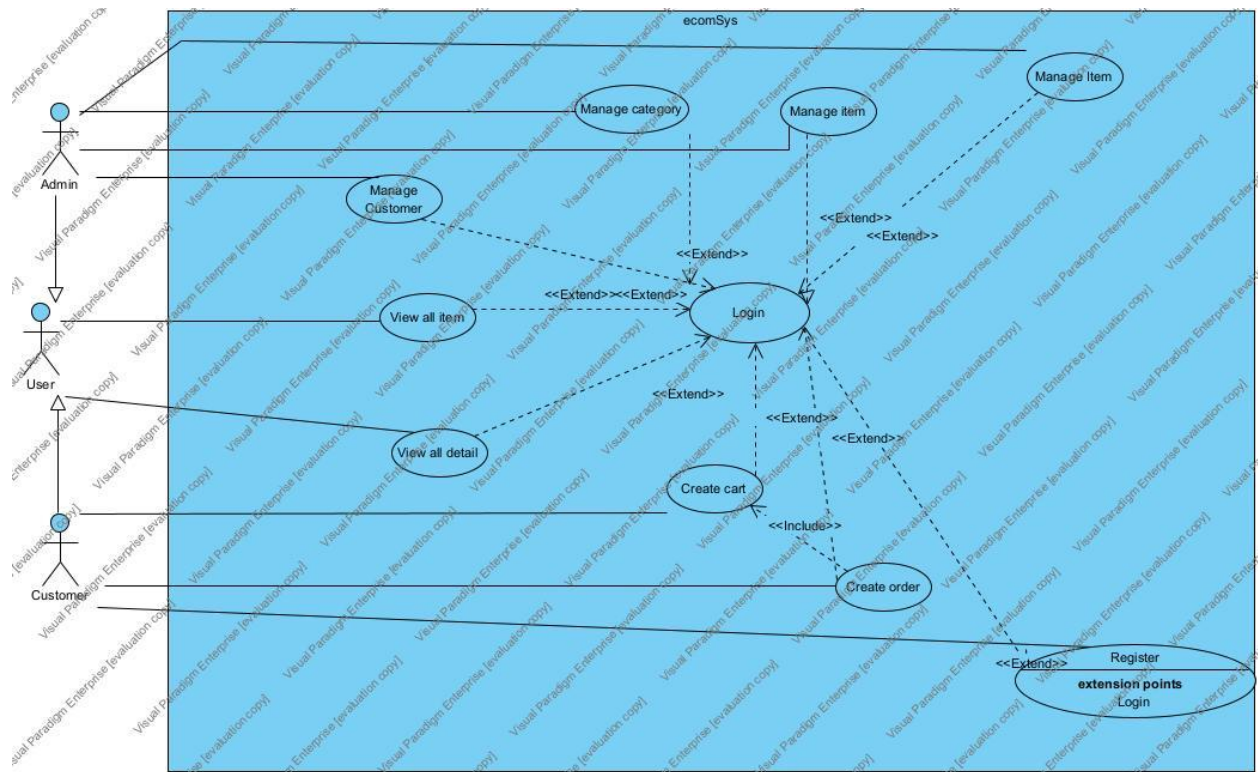
### 1.3. Chức năng cho Customer

Chức năng được thực hiện bởi người dùng Khách hàng:

- Đăng ký: Khách hàng có thể đăng ký trên website bằng số điện thoại và email.
- Đăng nhập khách hàng : Đây là form đăng nhập, từ đó khách hàng có thể đăng nhập vào hệ thống
- Customer Cart: Đây là giỏ hàng của khách hàng.
  - + Thêm sản phẩm vào giỏ hàng
  - + Chỉnh sửa giỏ hàng
  - + Xem giỏ hàng
- Customer Order: Đơn hàng của khách hàng Tạo đơn hàng mới: Chọn hình thức vận chuyển, Chọn hình thức thanh toán
- Phản hồi cho sản phẩm đã mua: Đây là mẫu phản hồi của khách hàng để khách hàng có thể đưa ra những phản hồi về sản phẩm sau khi đã nhận được hàng
- Đổi mật khẩu: Đây là module đổi mật khẩu để khách hàng đổi mật khẩu tài khoản của mình.
- Xem danh sách tất cả các sản phẩm
- Tìm kiếm sản phẩm

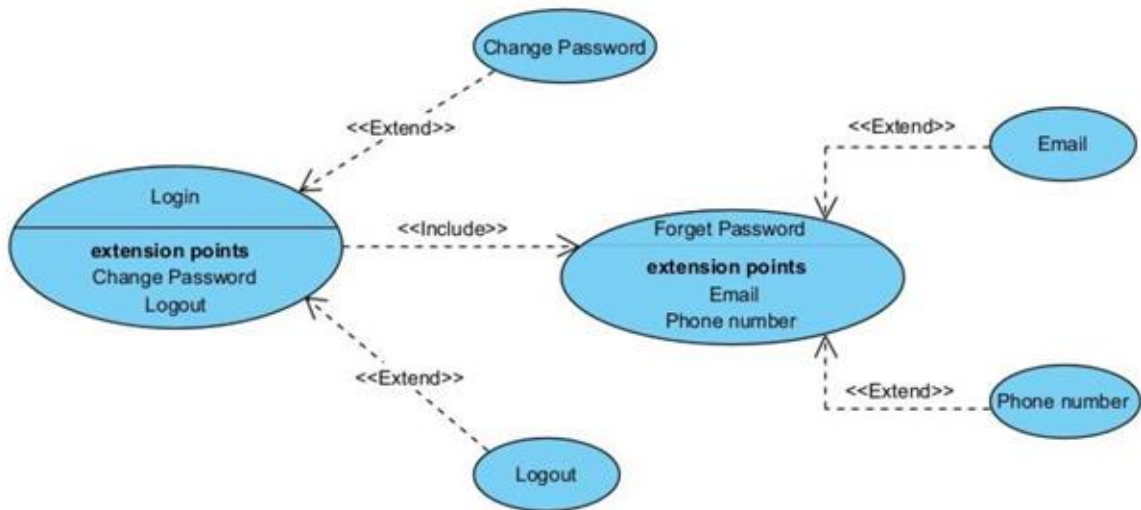
### Câu 2: Use case

Use case tổng Quát

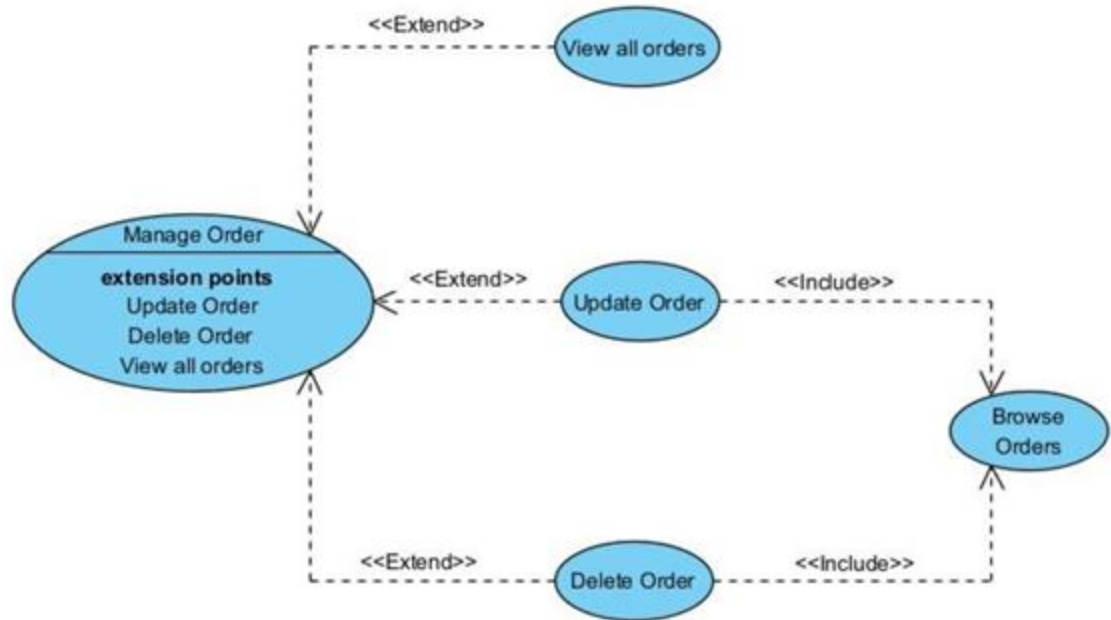


Use case chi tiet:

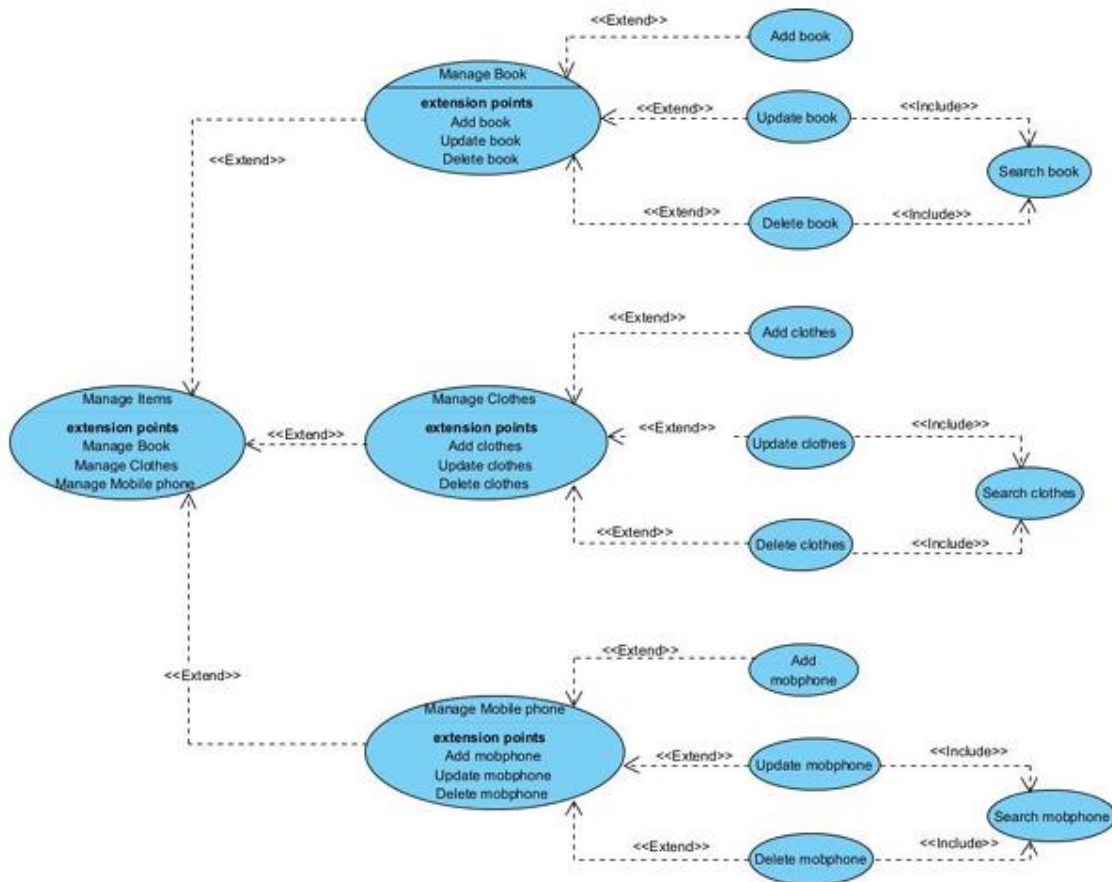
### 1. Login



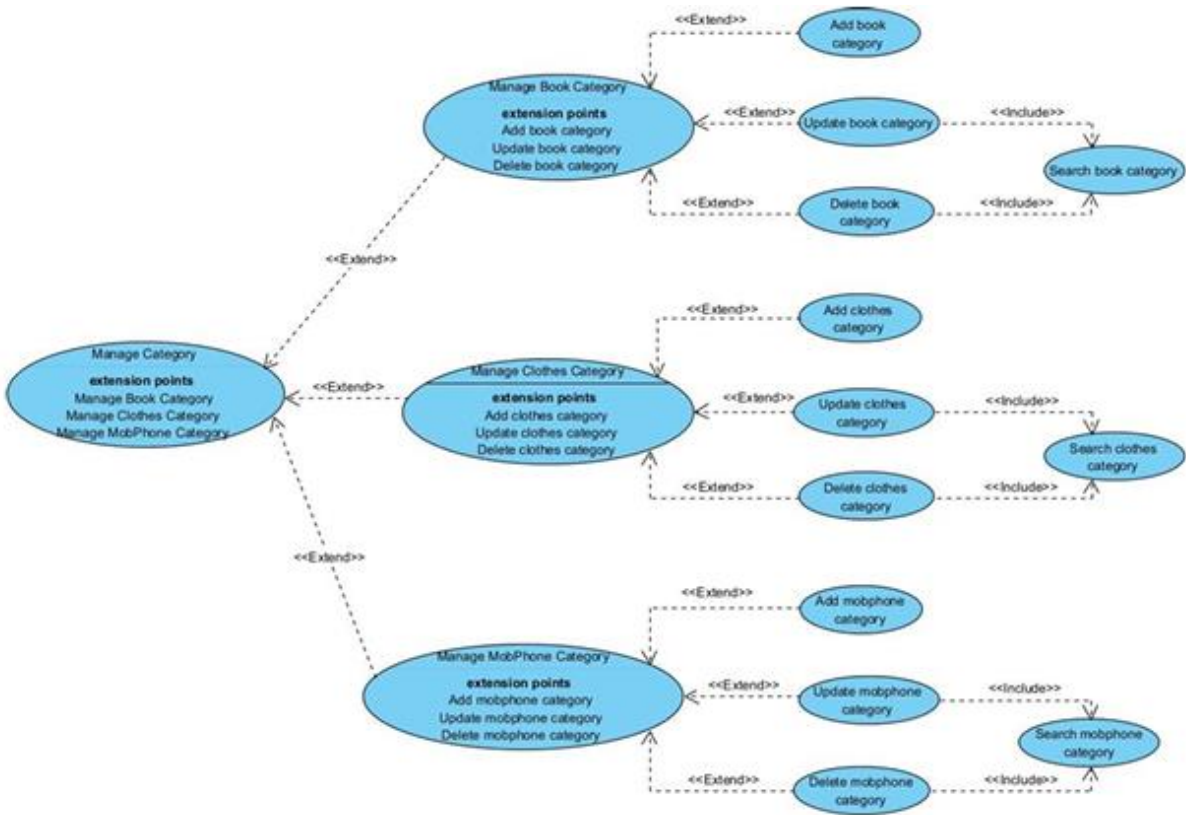
### 2. Manage Order



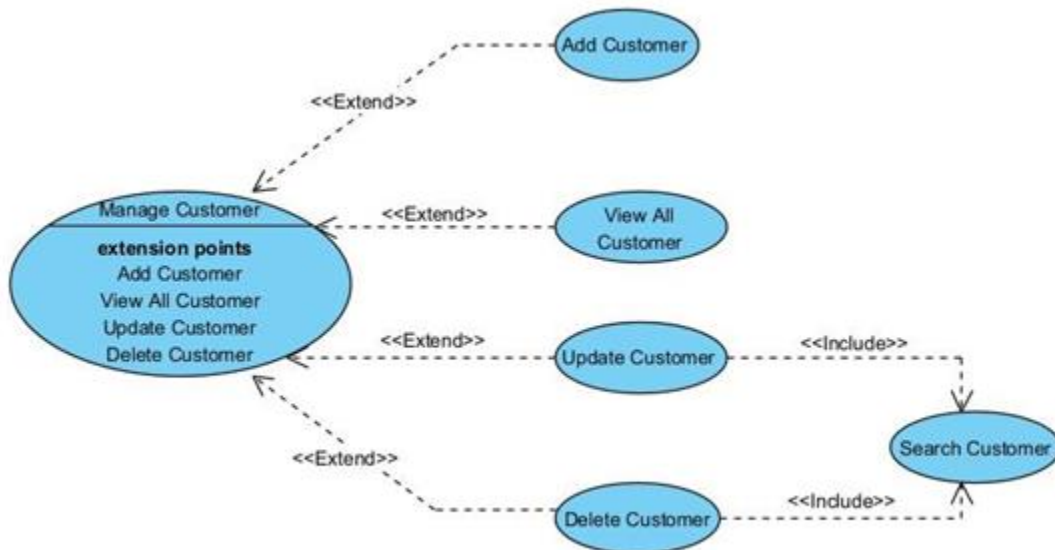
### 3. Manage Item



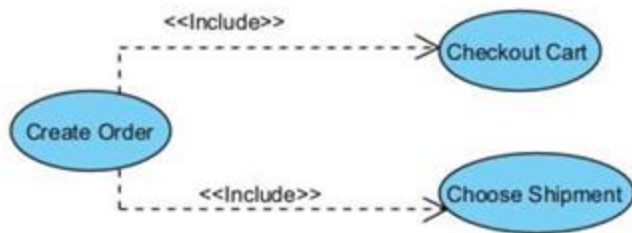
#### 4. Manage Category



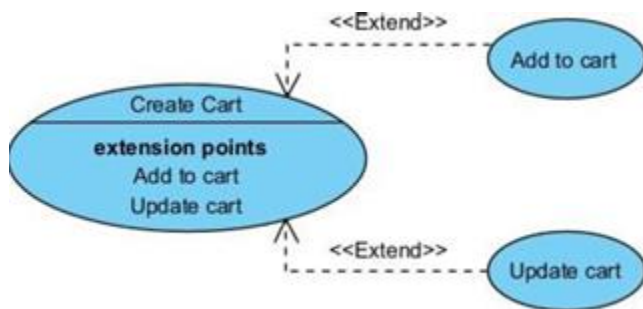
#### 5. Manage Customer



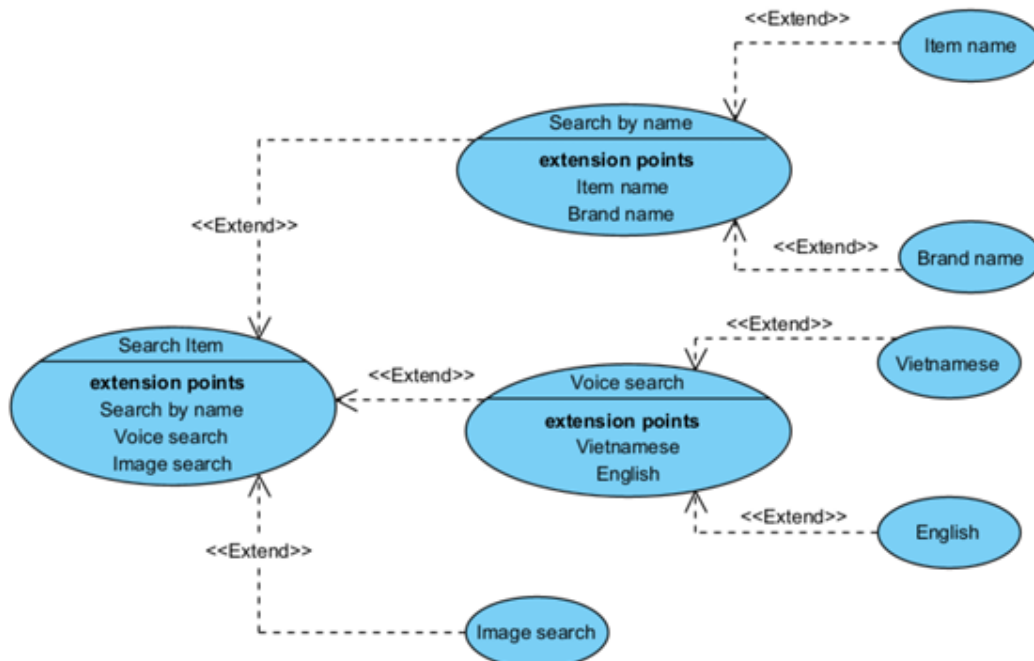
#### 6. Create Order



## 7. Create Cart



## 8. Search Item

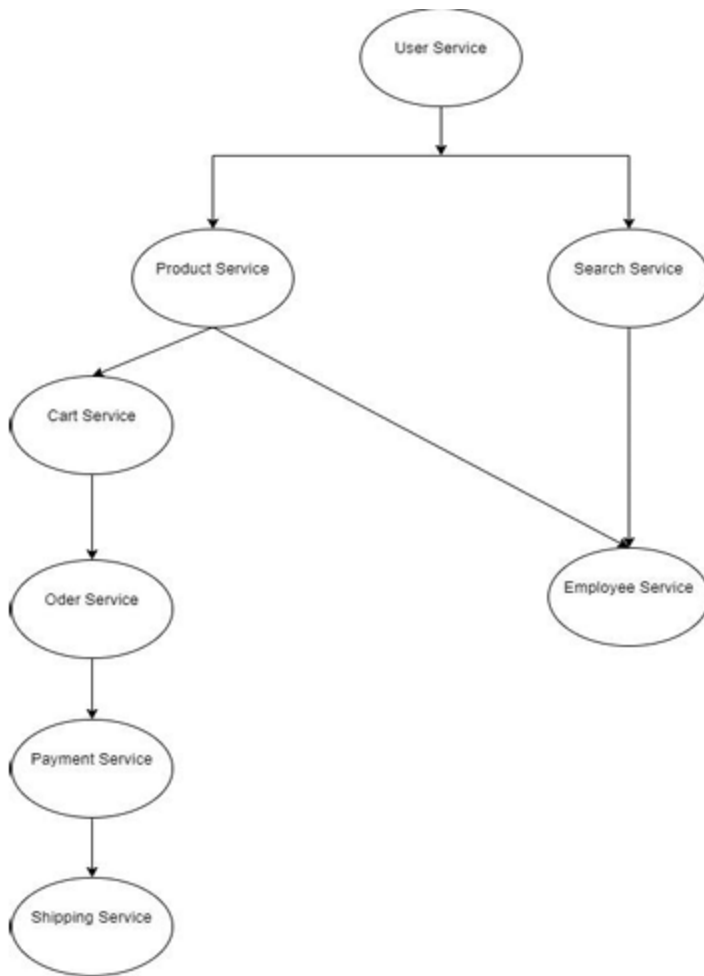


Câu 3: Vẽ biểu đồ phân rã Hệ ecomSys thành các service và các tương tác giữa các dịch vụ (sử dụng quan hệ <> trong UML).

❖ **Hệ ecomSys sẽ được phân rã thành các service như sau:**

- Dịch vụ người dùng: Dịch vụ này liên quan đến việc quản lý tài khoản người dùng và xác thực, chẳng hạn như cho phép người dùng tạo và đăng nhập vào tài khoản của họ cũng như lưu trữ thông tin hồ sơ người dùng.
- Dịch vụ sản phẩm: Dịch vụ này liên quan đến việc quản lý sản phẩm có sẵn để bán trên trang web, chẳng hạn như thêm sản phẩm mới, cập nhật sản phẩm, xoá sản phẩm hay xem chi tiết sản phẩm
- Dịch vụ tìm kiếm: Dịch vụ này liên quan đến việc tìm kiếm các sản phẩm trên trang web để xem chi tiết, thêm giỏ hàng hoặc mua sắm. Có thể tìm kiếm bằng text.
- Dịch vụ giỏ hàng: Dịch vụ này có khả năng chịu trách nhiệm quản lý các chức năng giỏ hàng của trang web, chẳng hạn như cho phép người dùng thêm các mặt hàng vào giỏ hàng của họ, cập nhật giỏ hàng khi các mặt hàng được thêm vào hoặc xoá và tính toán tổng chi phí của đơn hàng.
- Dịch vụ đặt hàng: Dịch vụ này liên quan đến quản lý quy trình đặt hàng và thực hiện đơn hàng, chẳng hạn như nhận và xử lý đơn hàng, theo dõi mức tồn kho và cập nhật cho khách hàng về trạng thái đơn hàng.
- Dịch vụ thanh toán: Dịch vụ này sẽ xử lý quá trình thanh toán, chẳng hạn như chấp nhận thanh toán bằng tài khoản online (ví điện tử), xác minh tính hợp lệ của thông tin thanh toán và bắt đầu giao dịch với cổng thanh toán hoặc ngân hàng.
- Dịch vụ vận chuyển: Dịch vụ này sẽ quản lý việc vận chuyển và giao sản phẩm cho khách hàng, chẳng hạn như theo dõi các gói hàng, phối hợp với hãng vận chuyển và tính toán chi phí vận chuyển.
- Dịch vụ nhân viên: Dịch vụ này có thể liên quan đến việc quản lý tài khoản và quyền của nhân viên, chẳng hạn như cho phép nhân viên truy cập vào một số phần nhất định của trang web và theo dõi chỉ số hiệu suất của nhân viên.

❖ **Biểu đồ phân rã**



Câu 4:

❖ Các dạng communication giữa các service synchronous:

- HTTP/HTTPS Request-Response: Trong giao thức HTTP, dịch vụ gửi một yêu cầu (request) đến một dịch vụ khác và đợi cho đến khi nhận được một phản hồi (response) trước khi tiếp tục thực hiện. Quá trình này là đồng bộ vì dịch vụ gửi yêu cầu phải chờ cho đến khi nhận được phản hồi.

Ví dụ:

```
import requests

def make_request():
    response = requests.get('https://api.example.com/data')
    return response.text
```

- Remote Procedure Call (RPC): RPC là một phương thức giao tiếp mà một dịch vụ gửi một yêu cầu cho một dịch vụ khác để thực thi một hàm hoặc phương thức cụ thể và chờ đợi kết quả



trước khi tiếp tục thực hiện. Quá trình này cũng là đồng bộ vì dịch vụ gọi trực tiếp chờ đợi cho đến khi hàm hoặc phương thức được thực thi và kết quả được trả về.

Ví dụ:

```
import xmlrpc.client

def rpc_call():
    proxy = xmlrpc.client.ServerProxy("http://localhost:8000/")
    result = proxy.add(4, 5)
    return result
```

- WebSocket Communication: Trong môi trường WebSocket, dịch vụ có thể thiết lập một kết nối hai chiều và gửi và nhận dữ liệu một cách liên tục. Mặc dù WebSocket cho phép giao tiếp real-time, nhưng quá trình gửi và nhận dữ liệu vẫn là đồng bộ.

Ví dụ:

```
import websocket

def websocket_communication():
    ws = websocket.create_connection("ws://localhost:8000/")
    ws.send("Hello, server!")
    result = ws.recv()
    ws.close()
    return result
```

❖ Các dạng communication giữa các service asynchronous:

- Message Queues: Trong hệ thống hàng đợi tin nhắn, dịch vụ gửi tin nhắn đến một hàng đợi mà không cần chờ đợi cho đến khi tin nhắn được xử lý. Dịch vụ khác có thể nhận và xử lý các tin nhắn từ hàng đợi một cách không đồng bộ, có nghĩa là chúng có thể tiếp tục thực hiện các công việc khác trong khi đang chờ đợi tin nhắn.

Ví dụ:

```

import pika

def send_message_to_queue():
    connection = pika.BlockingConnection(pika.ConnectionParameters('localhost'))
    channel = connection.channel()
    channel.queue_declare(queue='task_queue')
    channel.basic_publish(exchange='',
                          routing_key='task_queue',
                          body='Hello, queue!')
    connection.close()

def receive_message_from_queue():
    connection = pika.BlockingConnection(pika.ConnectionParameters('localhost'))
    channel = connection.channel()
    method_frame, header_frame, body = channel.basic_get(queue='task_queue')
    if method_frame:
        channel.basic_ack(method_frame.delivery_tag)
        return body.decode('utf-8')
    else:
        return None
    connection.close()

```

- Event Streams: Trong một môi trường dữ liệu dòng sự kiện, dịch vụ có thể gửi và nhận các sự kiện một cách không đồng bộ. Các sự kiện có thể được gửi và xử lý ngay khi chúng xảy ra mà không cần chờ đợi cho đến khi dịch vụ khác phản hồi.

Ví dụ:

```

import asyncio

async def event_stream_consumer():
    while True:
        event = await get_event_from_stream()
        process_event(event)

async def get_event_from_stream():
    # Code to fetch event from stream asynchronously
    pass

def process_event(event):
    # Code to process event
    pass

```

- Callback Functions: Trong một số trường hợp, các dịch vụ có thể sử dụng callback functions để thực hiện giao tiếp không đồng bộ. Dịch vụ gửi yêu cầu và chỉ định một hàm callback để xử lý kết quả khi nó được trả về.

Ví dụ:

```

import requests

def make_async_request(callback):
    response = requests.get('https://api.example.com/data')
    callback(response.text)

def process_response(response):
    # Code to process response
    pass

make_async_request(process_response)

```

Câu 5:

Giao tiếp đồng bộ:

- Payment service: cần đợi người dùng xác nhận thanh toán và xác thực được người dùng mới bắt đầu thanh toán. Và cần đợi hệ thống xử lý thanh toán xong mới có thể lấy được trạng thái thanh toán và update trạng thái thanh toán
- Search service: cần đợi các api search book của service book, search clothes của service clothes, search mobile của service mobile trả về kết quả mới có sản phẩm hiển thị lên cho người dùng
- User service: Khi user đăng nhập, sau khi gọi api login, cần đợi trả về kết quả mới quyết định báo lỗi hay thông báo đăng nhập thành công và hiển thị số lượng sản phẩm trong giỏ hàng
- Book service: khi thêm sách mới, sau khi gọi api để thêm sách, cần đợi kết quả trả về rồi quyết định hiển thị thông báo lỗi thêm sách hay thêm sách thành công. Tìm sách cũng tương tự như vậy, xem chi tiết sách cũng tương tự như vậy.
- Tương tự với Clothes service và Mobile Service
- Cart service:
  - + Khi gọi api của book, clothes, mobile để lấy thông tin sản phẩm thì cần đợi các api này trả về kết quả rồi mới có thể hiển thị lên giao diện của người dùng

Giao tiếp không đồng bộ:

- Order service: Sau khi người dùng đặt hàng, đơn hàng sẽ được chuyển sang cho người bán xác nhận. Lúc này người dùng có thể tiếp tục thực hiện các hành động khác mà không cần đợi người bán xác nhận đơn hàng